

#### REMARKS/ARGUMENTS

Claims 1-39 are pending in the application. Claim 1 has been amended to correct a minor error. Reconsideration is respectfully requested. Applicants submit that the pending claims 1-39 are patentable over the art of record and allowance is respectfully requested of claims 1-39.

Applicants would like to thank Examiner Timblin for holding a telephone interview with their representative, Janaki K. Davda, on Monday, April 9, 2007, at 2:30 p.m. (EST). Claims 1, 6, and 12 and the Andrei patent are to be discussed. No agreement was reached.

Claim 1 is objected to. Applicants have amended claim 1 to overcome the objection.

Claims 1-39 are rejected under 35 U.S.C. 102(e) as being anticipated by Andrei et al. (U.S. Patent No. 6,801,905 B2). Applicants respectfully traverse.

Anticipation requires that the identical invention must be shown in a single reference in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989).

Claims 1, 14, and 27 describe, under control of an iterator function processor that executes as part of a data store engine (e.g., Specification, page 6, paragraph 18, FIG. 1), when an iterator function included in a statement is invoked, obtaining one or more predicates included in the statement, applying the one or more predicates to a row of data, if applying the one or more predicates results in a match, returning the row of data, and, if applying the one or more predicates does not result in a match, searching for another row of data for which application of the one or more predicates results in a match.

On the other hand, as described in Applicants' Background of the Invention, a conventional iterator function receives a set of arguments and returns a table to the SQL statement that invokes the function (e.g., Specification, page 1, paragraph 4). In particular, the iterator function creates a virtual table with a result set, and then the qualification (i.e., predicate) is applied to the virtual table to filter rows of data in the virtual table (e.g., Specification, page 3, paragraph 9). In many cases, only a small percentage of rows of data in the virtual table remain

after the qualification is applied, but, unfortunately, because the qualification is applied after the rows of data are retrieved for the result set, many rows of data are unnecessarily retrieved for the virtual table (e.g., Specification, page 3, paragraph 9). For example, an application program submitting SQL statement (2) is interested in articles with a score (assigned by the text\_search() iterator function) greater than or equal to 0.9, but, because the text\_search() iterator function is not aware of the score qualification, the text\_search() iterator function returns a virtual table with all articles that have "Bush" in the same sentence as "recession," without applying the predicate for the score (e.g., Specification, page 3, paragraphs 10-11).

To avoid this problem, under control of an iterator function processor, when an iterator function included in a statement is invoked, one or more predicates included in the statement are obtained and the one or more predicates are applied to a row of data before the row of data is returned.

With reference to "when an iterator function included in a statement is invoked", the Examiner cites figure 8A, Col. 11, lines 4-13, and Col. 24, lines 25-67. Figure 8A describes receiving a query that is parsed and normalized and describes adding subplans to an equivalence class. Applicants respectfully submit that Figure 8A does not anticipate "when an iterator function included in a statement is invoked". Col. 11, lines 4-13, describe that SQL statements are passed to the parser which converts the statements into a query tree. Applicants respectfully submit that this does not anticipate "when an iterator function included in a statement is invoked".

At Col. 6, lines 23-27, the Andrei patent describes that a query execution plan or QEP is a demand-driven tuple stream "iterator" tree, which is a data structure that is interpreted by the relational database management system's execution module or engine. Col. 24, lines 25-67, describe that the physical operators module implements optimization time physical operators that are used for the optimization of the Volcano runtime iterators and describes that they are not themselves runtime iterators that implement the open/next/close protocol, that, rather, the physical operators module contains optimization time descriptions of those runtime iterators which are used for evaluation and costing during the optimization process. Applicants respectfully submit that mention of runtime iterators does not anticipate "when an iterator function included in a statement is invoked".

As to obtaining one or more predicates included in the statement, applying the one or more predicates to a row of data, if applying the one or more predicates results in a match, returning the row of data, and, if applying the one or more predicates does not result in a match, searching for another row of data for which application of the one or more predicates results in a match, Applicants respectfully submit that the Andrei patent does not describe that any of this processing occurs "when an iterator function included in a statement is invoked".

Also, the Andrei patent describes at Col. 11, lines 33-40, that, operating under the control of these instructions, the execution unit generates calls into lower-level routines, such as the access methods, for retrieving relevant information (e.g., row) from the database table, and, after the plan has been executed by the execution unit, the server returns a query result or answer table back to the client(s). Applicants respectfully submit that merely retrieving relevant information does not anticipate "when an iterator function included in a statement is invoked . . . if applying the one or more predicates results in a match, returning the row of data, and, if applying the one or more predicates does not result in a match, searching for another row of data for which application of the one or more predicates results in a match."

Thus, claims 1, 14, and 27 are not anticipated by the Andrei patent.

Dependent claims 2-5, 15-18, and 28-31 incorporate the language of independent claims 1, 14, and 27 and add additional novel elements. Therefore, dependent claims 2-5, 15-18, and 28-31 are not anticipated by the Andrei patent for at least the same reasons as were discussed with respect to claims 1, 14, and 27.

Also, claim 2 describes that, under control of an iterator function processor, when an iterator function included in a statement is invoked (from independent claim 1), obtaining the one or more predicates comprises obtaining a qualification descriptor that describes the one or more predicates and one or more functions that are used to process the one or more predicates. The Examiner cites figure 5 as anticipating this. Figure 5 of the Andrei patent mentions "applied predicates" and "predicates to apply", but figure 5 does not describe the one or more functions that are used to process the one or more predicates.

Claim 3 depends from claim 2 and describes that each function is used to process one of the predicates. The Examiner cites Col. 31, lines 1-11, as teaching this. The cited portion of the Andrei patent describes a sub-plan, but there is no description of each function being used to process one of the predicates.

Claim 5 describes that the iterator function is invoked by the data store engine and returns the row of data to the data store engine. The Examiner cites Col. 11, lines 33-40 as teaching this. The cited portion of the Andrei patent describes that the execution unit generates calls for retrieving relevant information and that after the plan has been executed by the execution unit, the server returns the query result or answer table back to the client. Neither the execution unit retrieving relevant information nor returning the query result to the client anticipates the iterator function returning a row of data to the data store engine.

Claims 6, 19, and 32 describe, under control of a data store engine, receiving a statement including an iterator function and one or more predicates, creating a qualification descriptor that describes the one or more predicates and one or more functions that are to be used to evaluate the one or more predicates, and invoking the iterator function one or more times, until receiving a done indicator from the iterator function, wherein each invocation of the iterator function results in receiving either a row of data for which at least one predicate has been applied or the done indicator and wherein one or more of the predicates included in the statement that have not been applied by the iterator function are applied by the data store engine.

The Examiner cites figure 5 as teaching creating a qualification descriptor that describes the one or more predicates and one or more functions that are to be used to evaluate the one or more predicates. Applicants respectfully submit that figure 5 does not describe one or more functions that are to be used to evaluate the one or more predicates.

The Examiner cites figures 5 and 8B and Col. 11, lines 33-40, as teaching invoking the iterator function one or more times, until receiving a done indicator from the iterator function, wherein each invocation of the iterator function results in receiving either a row of data for which at least one predicate has been applied or the done indicator and wherein one or more of the predicates included in the statement that have not been applied by the iterator function are applied by the data store engine. Applicants respectfully submit that figure 5 merely mentions "applied predicates", while figure 8B describes determining a final plan for execution of the query and generating a physical operator tree. Col. 11, lines 33-40 describe that, operating under the control of these instructions, the execution unit generates calls into lower-level routines, such as the access methods, for retrieving relevant information (e.g., row) from the database table, and, after the plan has been executed by the execution unit, the server returns a query result or

answer table back to the client(s). Applicants respectfully submit that the identical invention of invoking the iterator function one or more times until receiving a done indicator from the iterator function, wherein each invocation of the iterator function results in receiving either a row of data for which at least one predicate has been applied or the done indicator and wherein one or more of the predicates included in the statement that have not been applied by the iterator function are applied by the data store engine is not shown by the Andrei patent. Also, because the execution unit generates calls for retrieving information, there is no need for the data store engine to apply predicates included in the statement that have not been applied by the iterator function.

Thus, claims 6, 19, and 32 are not anticipated by the Andrei patent.

Dependent claims 7-11, 20-24, 33-37 incorporate the language of independent claims 6, 19, and 32 and add additional novel elements. Therefore, dependent claims 7-11, 20-24, 33-37 are not anticipated by the Andrei patent for at least the same reasons as were discussed with respect to claims 6, 19, and 32.

Claims 12, 25, and 38 describe the interaction between a data store engine and an iterator function processor in processing an iteration function. Under control of a data store engine, a statement including an iterator function and one or more predicates is received, a qualification descriptor is created that describes the one or more predicates and one or more functions that are to be used to evaluate the one or more predicates, and the iterator function is invoked. Under control of an iterator function processor, the qualification descriptor is retrieved, a row of data that matches at least one of the one or more predicates in the qualification descriptor is obtained; indicators are set to indicate which of the one or more predicates have been applied by the iterator function processor; and the row of data is returned to the data store engine. Under control of the data store engine, any of the one or more of the predicates included in the statement that have not been applied by the iterator function processor are applied.

As to, under control of a data store engine a qualification descriptor is created that describes the one or more predicates and one or more functions that are to be used to evaluate the one or more predicates, figure 5 does not describe one or more functions that are to be used to evaluate the one or more predicates.

As to, under control of an iterator function processor, the qualification descriptor is retrieved, figure 5 does not show the identical invention of a qualification descriptor that

describes the one or more predicates and one or more functions that are to be used to evaluate the one or more predicates as the one or more functions are not described. As to, under control of the data store engine, any of the one or more of the predicates included in the statement that have not been applied by the iterator function processor are applied, because the execution unit of the Andrei patent generates calls for retrieving information, there is no need for the data store engine to apply predicates included in the statement that have not been applied by the iterator function.

Applicants respectfully submit that the Andrei patent does not anticipate the subject matter of claims 12, 25, and 38.

Dependent claims 13, 26, and 39 incorporate the language of independent claims 12, 25, and 38 and add additional novel elements. Therefore, dependent claims 13, 26, and 39 are not anticipated by the Andrei patent for at least the same reasons as were discussed with respect to claims 12, 25, and 38.

#### Conclusion

For all the above reasons, Applicants submit that the pending claims 1-39 are patentable over the art of record. Applicants have not added any claims. Nonetheless, should any additional fees be required, please charge Deposit Account No. 09-0460.

The attorney of record invites the Examiner to contact her at (310) 553-7973 if the Examiner believes such contact would advance the prosecution of the case.

Dated: April 9, 2007

By:    /Janaki K. Davda                    

Janaki K. Davda  
Registration No. 40,684

Please direct all correspondences to:

David Victor  
Konrad Raynes & Victor, LLP  
315 South Beverly Drive, Ste. 210  
Beverly Hills, CA 90212  
Tel: 310-553-7977  
Fax: 310-556-7984